UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/847,068	05/01/2001	Yoav Almog	42390P10913	6986

8791 7590 02/20/2004

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD, SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER
GOLE, AMOL V

ART UNIT	PAPER NUMBER
2183	2

DATE MAILED: 02/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/847,068	**Applicant(s)** ALMOG ET AL.
	Examin r Amol V. Gole	**Art Unit** 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 May 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 May 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been examined.

Specification

2. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

3. The abstract of the disclosure is objected to because its grammatical structure is very “claim-like” and does not provide a clear-cut insight on the invention. Correction is required. See MPEP § 608.01(b).

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: SPECULATIVE BRANCH TARGET ALLOCATION USING A SEPARATE CACHE FOR STORING SPECULATIVE TARGETS.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

6. Claims **1, 4-8, 11-14, and 17-19** are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung (US005774710A) in view of Tran et al. (US005878255A).

7. **In regard to claim 1:**

8. Chung discloses a method comprising:

determining a target of a branch instruction (col. 4, lines 50-52; Although not explicitly mentioned, because the decoder sends the branch targets to the branch prediction control logic, the target must be inherently determined in either the decode or fetch stages);

storing the target of the branch instruction (col. 4, lines 50-52, 39-41; the target sent from the decoder is stored to the branch target buffer 125); and

re-encountering the branch instruction and predicting a target for the branch instruction by accessing the stored target for the branch instruction (col. 4, lines 13-16, 20-23).

9. Chung differs from the instant invention because he does not mention that the target of the branch instruction is stored before the branch is fully executed.

10. Tran et al. teach the use of an update unit (fig. 2, branch holding register 250, branch update data register 256, comparator 253, and input/output mux 254,257) which allows branch prediction information, including branch targets (successor index col. 10, lines 21-27), to be updated speculatively i.e. before the branch is fully executed (col. 2, lines 63-64). Only when the branch instruction is fully executed i.e. becomes non-speculative is the update data written to the branch prediction array 255 (col. 11, lines

15-17). They further teach that if speculative updates were written directly into the array 255, then the branch information would be skewed with incorrect data if the speculation were wrong resulting in worsened prediction (col. 3, lines 8-12). Performance may be increased by speculatively updating branch prediction information (col. 3, lines 18-20). Moreover, the use of this update unit results in savings in cost and die area (col. 2, lines 49-50).

11. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Chung's branch prediction mechanism by adding an update unit like that of Tran et al.. The Branch Target Buffer 125 of Chung would be in the place of the branch prediction array 255 of Tran. Thus, targets sent from the decode stage could be stored speculatively in the branch update data register 256 of Tran, i.e. before the branch is fully executed.

12. One would have been motivated to make this modification because it allows for better performance, prediction and savings in cost leading to an efficient branch prediction mechanism which is the objective of the Chung invention (col. 2, lines 22-24).

13. In regard to claim 4:

14. The combination of Chung in view of Tran et al. further discloses in the method of claim 1, wherein storing the target comprises saving the target to a cache (Tran: branch update data register 256).

15. In regard to claim 5:

16. The combination of Chung in view of Tran et al. discloses the method of claim 4, wherein the target of the branch instruction is also stored in a branch prediction unit (Chung: Branch Target Buffer 125) after the branch instruction has been fully executed (Chung: fig. 2, col. 4, lines 52-55, 39-43 indicates that the branch target is sent to be stored to the branch prediction unit after execution).

17. In regard to claim 6:

18. The combination of Chung in view of Tran et al. discloses the method of claim 5, wherein the target is predicted for the branch instruction before the target of the branch instruction is stored in the branch prediction unit (Chung: col. 4, lines 20-23 shows that the fetch logic 110 uses the predicted target for fetching before the target is stored in the branch target buffer 125 [col. 4, lines 52-55, 39-43]).

19. In regard to claim 7:

20. The combination of Chung in view of Tran et al. teaches that predicting a target for a branch instruction comprises:

accessing at least one target stored in at least one of the cache (Tran: fig. 2, 256) and the branch prediction unit (Chung: fig. 1, 125);

prioritizing the accessed targets; and

generating a branch prediction based on the prioritized targets.

(Tran et al. teaches that if the currently fetched instruction's address is found in the branch holding register, the branch target (successor index col. 10, lines 21-27) is in the cache (branch update data register 256 col. 10, lines 21-26) and that target is used to generate prediction, otherwise the branch prediction array 255 (Chung: fig. 1, Branch Target Buffer 125) is accessed (col. 9, lines 19-25). Hence, Tran discloses a priority of first accessing the cache 256, then the branch prediction array 255.)

21. In regard to claim 8:

22. Chung discloses an apparatus comprising:

a decoder to determine a target of a branch instruction (Although not explicitly mentioned this limitation is deemed inherent because as the decode stage is the first stage to realize whether or not the fetched instruction is a branch or not and as Chung discloses that the decoder sends the target to the branch target buffer 125 (col. 4, lines 50-53, 39-43) the target must be determined by the decoder);

a cache (fig. 1, branch target buffer 125) to store the target of the branch instruction (col. 4, lines 50-52, 39-41; the target sent from the decoder is stored to the branch target buffer 125); and

a branch prediction unit (fig. 1, branch target buffer 125 and branch prediction control 135) to, upon re-encountering the branch instruction, predict the target of the branch instruction by accessing the target of the branch instruction stored in the cache (col. 4, lines 13-16, 20-23).

23. Chung differs from the instant invention because he does not mention that the target of the branch instruction is stored in a cache before the branch is fully executed.

24. Tran et al. teach the use of an update unit (fig. 2, branch holding register 250, branch update data register 256, comparator 253, and input/output mux 254,257), which allows branch prediction information, including branch targets (successor index col. 10, lines 21-27), to be updated speculatively i.e. before the branch is fully executed (col. 2, lines 63-64). Only when the branch instruction is fully executed i.e. becomes non-speculative is the update data written to the branch prediction array 255 (col. 11, lines 15-17). They further teach that if speculative updates were written directly into the array 255, then the branch information would be skewed with incorrect data if the speculation were wrong resulting in worsened prediction (col. 3, lines 8-12). Performance may be increased by speculatively updating branch prediction information (col. 3, lines 18-20). Moreover, the use of this update unit results in savings in cost and die area (col. 2, lines 49-50).

25. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Chung's branch prediction mechanism by adding an update unit like that of Tran et al.. The Branch Target Buffer 125 of Chung would be in the place of the branch prediction array 255 of Tran and the branch update data register 256 of Tran would be the cache. Thus, targets sent from the decode stage could be stored speculatively in the branch update data register 256 of Tran, i.e. before the branch is fully executed.

26. One would have been motivated to make this modification because it allows for better performance, prediction and savings in cost leading to an efficient branch prediction mechanism which is the objective of the Chung invention (col. 2, lines 22-24).

27. In regard to claim 11:

28. The combination of Chung in view of Tran et al. discloses the apparatus of claim 8, wherein the branch prediction unit (Chung: Branch Target Buffer 125) also stores the target of the branch instruction after the branch instruction has been fully executed (Chung: fig. 2, col. 4, lines 52-55, 39-43 indicates that the branch target is sent to be stored to the branch prediction unit after execution).

29. In regard to claim 12:

30. The combination of Chung in view of Tran et al. discloses the apparatus of claim 11, the branch prediction unit predicts the target for the branch instruction before the target of the branch instruction is stored in the branch prediction unit (Chung: col. 4, lines 20-23 shows that the fetch logic 110 uses the predicted target for fetching before the target is stored in the branch target buffer 125 [col. 4, lines 52-55, 39-43]).

31. In regard to claim 13:

32. The combination of Chung in view of Tran et al. teaches that the branch prediction unit predicts the target for the branch instruction by:

accessing at least one target stored in at least one of the cache (Tran: fig. 2, 256) and the branch prediction unit (Chung: fig. 1, 125);

prioritizing the accessed targets; and

generating a branch prediction based on the prioritized targets.

(Tran et al. teaches that if the currently fetched instruction's address is found in the branch holding register, the branch target (successor index col. 10, lines 21-27) is in the cache (branch update data register 256 col. 10, lines 21-26) and that target is used to generate prediction, otherwise the branch prediction array 255 (Chung: fig. 1, Branch Target Buffer 125) is accessed (col. 9, lines 19-25). Hence, Tran discloses a priority of first accessing the cache 256, then the branch prediction array 255.)

33. **In regard to claim 14:**

34. Chung discloses an system comprising:

a processor capable of pipelining instructions (fig. 1);

a decoder to determine a target of a branch instruction (Although not explicitly mentioned this limitation is deemed inherent because as the decode stage is the first stage to realize whether or not the fetched instruction is a branch or not and as Chung discloses that the decoder sends the target to the branch target buffer 125 (col. 4, lines 50-53, 39-43) the target must be determined by the decoder);

a cache (fig. 1, branch target buffer 125) to store the target of the branch instruction (col. 4, lines 50-52, 39-41; the target sent from the decoder is stored to the branch target buffer 125); and

a branch prediction unit (fig. 1, branch target buffer 125 and branch prediction control 135) to, upon re-encountering the branch instruction, predict the target of the branch instruction by accessing the target of the branch instruction stored in the cache (col. 4, lines 13-16, 20-23).

35. Chung differs from the instant invention because he does not mention that the target of the branch instruction is stored in a cache before the branch is fully executed.

36. Tran et al. teach the use of an update unit (fig. 2, branch holding register 250, branch update data register 256, comparator 253, and input/output mux 254,257), which allows branch prediction information, including branch targets (successor index col. 10, lines 21-27), to be updated speculatively i.e. before the branch is fully executed (col. 2, lines 63-64). Only when the branch instruction is fully executed i.e. becomes non-

speculative is the update data written to the branch prediction array 255 (col. 11, lines 15-17). They further teach that if speculative updates were written directly into the array 255, then the branch information would be skewed with incorrect data if the speculation were wrong resulting in worsened prediction (col. 3, lines 8-12). Performance may be increased by speculatively updating branch prediction information (col. 3, lines 18-20). Moreover, the use of this update unit results in savings in cost and die area (col. 2, lines 49-50).

37. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Chung's branch prediction mechanism by adding an update unit like that of Tran et al.. The Branch Target Buffer 125 of Chung would be in the place of the branch prediction array 255 of Tran and the branch update data register 256 of Tran would be the cache. Thus, targets sent from the decode stage could be stored speculatively in the branch update data register 256 of Tran, i.e. before the branch is fully executed.

38. One would have been motivated to make this modification because it allows for better performance, prediction and savings in cost leading to an efficient branch prediction mechanism which is the objective of the Chung invention (col. 2, lines 22-24).

39. In regard to claim 17:

40. The combination of Chung in view of Tran et al. discloses the system of claim 14, wherein the branch prediction unit (Chung: Branch Target Buffer 125) also stores the target of the branch instruction after the branch instruction has been fully executed (Chung: fig. 2, col. 4, lines 52-55, 39-43 indicates that the branch target is sent to be stored to the branch prediction unit after execution).

41. In regard to claim 18:

42. The combination of Chung in view of Tran et al. discloses the system of claim 17, wherein the branch prediction unit predicts the target for the branch instruction before the target of the branch instruction is stored in the branch prediction unit (Chung: col. 4, lines 20-23 shows that the fetch logic 110 uses the predicted target for fetching before the target is stored in the branch target buffer 125 [col. 4, lines 52-55, 39-43]).

43. In regard to claim 19:

44. The combination of Chung in view of Tran et al. teaches that the branch prediction unit predicts the target for the branch instruction by:

accessing at least one target stored in at least one of the cache (Tran: fig. 2, 256) and the branch prediction unit (Chung: fig. 1, 125);

prioritizing the accessed targets; and

generating a branch prediction based on the prioritized targets.

(Tran et al. teaches that if the currently fetched instruction's address is found in the branch holding register, the branch target (successor index col. 10, lines 21-27) is in the cache (branch update data register 256 col. 10, lines 21-26) and that target is used to generate prediction, otherwise the branch prediction array 255 (Chung: fig. 1, Branch Target Buffer 125) is accessed (col. 9, lines 19-25). Hence, Tran discloses a priority of first accessing the cache 256, then the branch prediction array 255.)

45. Claims **2, 3, 9, 10, 15 and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung (US005774710A) in view Tran et al. (US005878255A) in further view of Rappoport et al. (US006601161B2).

46. In regard to claim 2:

47. Chung in view Tran et al. does not explicitly mention that the branch instruction is a direct branch.

48. Rappoport et al. teach that the target of a direct branch is known after the decode stage (col. 12, lines 34-36).

49. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to store the target of a branch instruction that is a direct branch.

50. One would have been motivated to realize this because Chung mentions that the decode stage sends the branch target to be stored (col. 4, lines 50-52, 39-43) and Rappoport teaches that the target of a direct branch is known after the decode stage hence the target of a direct branch is to be stored.

51. In regard to claim 3:

52. Although Chung in view Tran et al. in further view of Rappoport does not explicitly mention that the branch instruction is a backward branch, a direct branch jumping to a previous memory location is a backward branch and hence the limitation is deemed inherent.

53. In regard to claim 9:

54. Chung in view Tran et al. does not explicitly mention that the decoder determines the target of a direct branch instruction.

55. Rappoport et al. teach that the target of a direct branch is known after the decode stage (col. 12, lines 34-36).

56. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to store the target of a branch instruction that is a direct branch.

57. One would have been motivated to realize this because Chung mentions that the decode stage sends the branch target to be stored (col. 4, lines 50-52, 39-43) and Rappoport teaches that the target of a direct branch is known after the decode stage hence the target of a direct branch is to be stored.

58. In regard to claim 10:

59. Although Chung in view Tran et al. further in view of Rappoport does not explicitly mention that the decoder determines the target of a backward branch instruction, a direct branch jumping to a previous memory location is a backward branch and hence the limitation is deemed inherent.

60. In regard to claim 15:

61. Chung in view Tran et al. does not explicitly mention that the decoder determines the target of a direct branch instruction.

62. Rappoport et al. teach that the target of a direct branch is known after the decode stage (col. 12, lines 34-36).

63. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to store the target of a branch instruction that is a direct branch.

64. One would have been motivated to realize this because Chung mentions that the decode stage sends the branch target to be stored (col. 4, lines 50-52, 39-43) and Rappoport teaches that the target of a direct branch is known after the decode stage hence the target of a direct branch is to be stored.

65. In regard to claim 16:

66. Although Chung in view Tran et al. further in view of Rappoport does not explicitly mention that the decoder determines the target of a backward branch instruction, a direct branch jumping to a previous memory location is a backward branch and hence the limitation is deemed inherent.

Conclusion

67. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty, which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections. See 37 CFR § 1.111.

a. Koino (US006609194B1) teaches a branch speculative branch update scheme that improves efficiency in code with small loops.

b. Hara (US005737590) teaches a branch prediction scheme using limited BTB updates.

c. Hennessey and Patterson ("Computer Architecture, A Quantitative Approach," Morgan Kaufmann, pp. 271-278, 1996), teaches the functioning of a typical BTB.

d. Keller et al. (US006647490B2) teach a branch prediction technique and direct and indirect branches (col. 15, lines 26-50).

e. Col et al. (US006526502B1) teach a speculative branch history update scheme.

f. Henry et al. (US006609194B1) teach a branch prediction technique basd on branch type.

68. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amol V. Gole whose telephone number is 703-305-8888. The examiner can normally be reached on 9:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 703-305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AVG
amol.gole@uspto.gov

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100